Compressive Sensing

Dr.-Ing. Ali Bereyhi

Friedrich-Alexander University

ali.bereyhi@fau.de

Summer 2022

Where Are We?

We ended up with following results

Moral of Story

- *Most signals represented with finite numbers of entries*
- *These entries are presented by a vector which is*

  Either sparse or compressible
- *We could recover the signal perfectly from few samples*

Where Are We?

And following questions

- *How much we could reduce the number of samples?*
- *How could we recover the signal efficiently from the samples?*
- *How should we perform the sampling?*
- *Can we guarantee the perfect recovery of a signal?*

Now, we try to answer

Our First Try for Sparse Recovery

Sparse Recovery: *Optimal Approach*

We now want to solve the sparse recovery problem:

Sparse Recovery

For given $\mathbf{A}$ *and* $\boldsymbol{y}$, *we intend to find vector* $\boldsymbol{x}$, *such that*

$$\mathbf{A}\boldsymbol{x} = \boldsymbol{y}$$
$$\text{subject to } \|\boldsymbol{x}\|_0 \leq s$$

for some sparsity s

At this point, we do not care about the complexity

Our First Try for Sparse Recovery

Section 1: Optimal Sparse Recovery via ℓ_0-Norm Minimization

Sparse Recovery: *Optimal Approach*

Assume that

$$\boxed{\boldsymbol{x}^\star \text{ is the unique solution of sparse recovery problem}}$$

What does this mean?

Consider the underdetermined system of equations

$$\mathbf{A}\boldsymbol{x} = \boldsymbol{y}$$

Let $\mathcal{Z}$ be the set of all solutions. This means

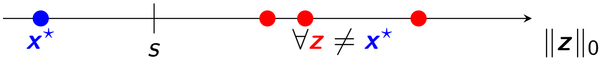
$$\forall \boldsymbol{z} \in \mathcal{Z} \rightsquigarrow \mathbf{A}\boldsymbol{z} = \boldsymbol{y} \quad \text{and} \quad \forall \boldsymbol{z} \notin \mathcal{Z} \rightsquigarrow \mathbf{A}\boldsymbol{z} \neq \boldsymbol{y}$$

Sparse Recovery: *Optimal Approach*

As a result, we can write

$$\boldsymbol{x}^\star \in \mathcal{Z}$$

What happens if we plot the ℓ_0-norm of all $\boldsymbol{z} \in \mathcal{Z}$?



- $\boldsymbol{x}^\star$ *is a solution of sparse recovery problem*

$$\|\boldsymbol{x}^\star\|_0 \leq s$$

- $\boldsymbol{x}^\star$ *is the unique solution*

$$\forall \mathcal{Z} \ni \boldsymbol{z} \neq \boldsymbol{x}^\star : \; \|\boldsymbol{z}\|_0 > s$$

Sparse Recovery: *Optimal Approach*

We can alternatively find the unique solution $\boldsymbol{x}^\star$ *as*

$$\boldsymbol{x}^\star = \operatorname*{argmin}_{\boldsymbol{z}\in\mathcal{Z}} \|\boldsymbol{z}\|_0$$

From the definition of $\mathcal{Z}$, *we rewrite this optimization as*

$$\boldsymbol{x}^\star = \operatorname{argmin} \|\boldsymbol{z}\|_0 \textit{ subject to } \mathbf{A}\boldsymbol{z} = \boldsymbol{y}$$

Clearly, $\boldsymbol{x}^\star$ *is also the unique solution of* ℓ_0*-norm minimization*

$$\boxed{\boldsymbol{x}^\star \textit{ solves sparse recovery} \leftrightarrows \boldsymbol{x}^\star \textit{ solves } \ell_0\textit{-norm minimization}}$$

Sparse Recovery: *Optimal Approach*

ℓ_0-Norm Recovery

The unique solution of the sparse recovery problem is given by

$$\min \|\boldsymbol{z}\|_0 \textit{ subject to } \mathbf{A}\boldsymbol{z} = \boldsymbol{y}$$

What does this mean?

If we have a unique solution for sparse recovery problem; then,

We can recover it for sure via the ℓ_0-norm recovery

We hence call ℓ_0-norm recovery the optimal recovery algorithm

Sparse Recovery: *Optimal Approach*

Now, we know the optimal recovery algorithm, but ...

How we could make sure that there exists a unique solution?

In other words,

What if there exists $\boldsymbol{x}^\sharp \neq \boldsymbol{x}^\star$, such that $\mathbf{A}\boldsymbol{x}^\sharp = \boldsymbol{y}$ and

$$\|\boldsymbol{x}^\sharp\|_0 \leq s?$$

If $\#$ of samples is larger than some $M^\star$, we can make sure!

We now find out what is $M^\star$

Summary

Up to now, we learned that ...

- *The optimal approach for sparse recovery is*

  ℓ_0-norm minimization

- *We need a minimum number of samples, to make sure*

  The sparse recovery is uniquely performed

But what is this minimal number of samples?

$\boxed{\textit{We now find it out for the optimal algorithm}}$

Minimal Number of Samples

Minimal Samples for Sparse Recovery

Depending on the information we have on the sparse signal

$$s \leq M^{\star} \leq 2s$$

We now find out how and when these criteria hold?

$$\boxed{\textit{We need to first go through some basic definitions}}$$

Some Basic Definitions: $\mathcal{S}$-Subvector

$\mathcal{S}$-Subvector

Let $\boldsymbol{x} \in \mathbb{R}^N$, and $\mathcal{S} \subset \{1, \dots, N\}$ of size L, i.e.,

$$\mathcal{S} = \{i_1, \dots, i_L\}$$

where $i_\ell \in \{1, \dots, N\}$ for $\ell \in \{1, \dots, L\}$. The $\mathcal{S}$-subvector of $\boldsymbol{x}$ is

$$\boldsymbol{x}_\mathcal{S} = \begin{bmatrix} x_{i_1} \\ \vdots \\ x_{i_L} \end{bmatrix} \in \mathbb{R}^L$$

Some Basic Definitions: $\mathcal{S}$-Subvector

Example: Consider vector $\boldsymbol{x}$

$$\boldsymbol{x} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 3 \end{bmatrix}$$

and let $\mathcal{S} = \{2, 4\}$. *The* $\mathcal{S}$*-subvector of* $\boldsymbol{x}$ *is*

$$\boldsymbol{x}_{\mathcal{S}} = \begin{bmatrix} 0 \\ 3 \end{bmatrix}$$

Some Basic Definitions: Column $\mathcal{S}$-Submatrix

Column $\mathcal{S}$-Submatrix

Let $\mathbf{A} = [\boldsymbol{a}_1, \dots, \boldsymbol{a}_N]$ *with* $\boldsymbol{a}_n \in \mathbb{R}^M$, *and*

$$\mathcal{S} = \{i_1, \dots, i_L\} \subset \{1, \dots, N\}$$

The column $\mathcal{S}$*-submatrix of* $\mathbf{A}$ *is*

$$\mathbf{A}_{\mathcal{S}} = [\boldsymbol{a}_{i_1}, \dots, \boldsymbol{a}_{i_L}] \in \mathbb{R}^{M \times L}$$

You can similarly define row $\mathcal{S}$*-submatrix*

Some Basic Definitions: Column $\mathcal{S}$-Submatrix

Example: Consider matrix $\mathbf{A}$

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 5 & 0 \\ 1 & 0 & 5 & 0 \\ 1 & 0 & 5 & 0 \end{bmatrix}$$

and let $\mathcal{S} = \{2, 3\}$*. The column* $\mathcal{S}$*-submatrix of* $\mathbf{A}$ *is*

$$\mathbf{A}_{\mathcal{S}} = \begin{bmatrix} 0 & 5 \\ 0 & 5 \\ 0 & 5 \end{bmatrix}$$

Some Basic Definitions: Column $\mathcal{S}$-Submatrix

Let's do an exercise together which also benefits us later on

Assume that $\text{Supp}\,(\boldsymbol{x}) = \mathcal{S}$; *show that*

$$\mathbf{A}\boldsymbol{x} = \mathbf{A}_{\mathcal{S}}\boldsymbol{x}_{\mathcal{S}}$$

We can try it first for

$$\boldsymbol{x} = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix} \qquad \textit{and} \qquad \mathbf{A} = [\boldsymbol{a}_1 \;\; \boldsymbol{a}_2 \;\; \boldsymbol{a}_3 \;\; \boldsymbol{a}_4]$$

Some Basic Definitions: Matrix Kernel

Matrix Kernel

Consider $\mathbf{A} \in \mathbb{R}^{M \times N}$. *The kernel or null space of* $\mathbf{A}$ *is*

$$\ker \mathbf{A} = \left\{ \textit{all } \boldsymbol{x} \in \mathbb{R}^N : \mathbf{A}\boldsymbol{x} = 0 \right\}$$

Two well-known facts about the kernel:

- *For any matrix* $\mathbf{A}$, *the vector of all zeros* $\mathbf{0} \in \ker \mathbf{A}$
- *The kernel of a fat matrix always has also other elements, i.e.,*

$$\ker \mathbf{A} \neq \{\mathbf{0}\}$$

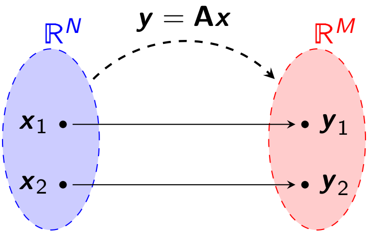
if $\#$ *rows of* $\mathbf{A} < \#$ *columns* $\mathbf{A}$

Some Basic Definitions: Injection

Injective Linear Mapping

Matrix $\mathbf{A} \in \mathbb{R}^{M \times N}$ *describes an injective linear mapping, if*

$$\forall \boldsymbol{x}_1 \neq \boldsymbol{x}_2 \in \mathbb{R}^N \rightsquigarrow \mathbf{A}\boldsymbol{x}_1 \neq \mathbf{A}\boldsymbol{x}_2$$

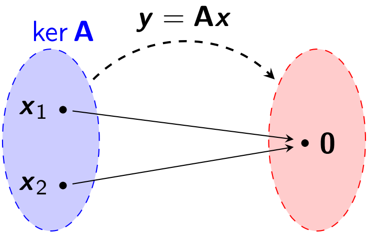


Some Basic Definitions: Injection

Necessary Condition for Injection

Let $\mathbf{A} \in \mathbb{R}^{M \times N}$ *describe an injective linear mapping; then,*

$$\ker \mathbf{A} = \{\mathbf{0}\}$$



Some Basic Definitions: Injection

Main Application of Injection

Let $\mathbf{A} \in \mathbb{R}^{M \times N}$ *describe an injective linear mapping; then, there exists a matrix* $\mathbf{B} \in \mathbb{R}^{N \times M}$

$$\mathbf{B}\mathbf{A} = \mathbf{I}_N$$

Roughly speaking,

If $\mathbf{A}$ *is injective; then, it is invertible*

Some Basic Definitions: Injection

For an square matrix, injection is exactly the invertibility

$\mathbf{A} \in \mathbb{R}^{N \times N}$ *is injective, if it is invertible, i.e.,* $\det \mathbf{A} \neq 0$

In this case, we simply have

$$\mathbf{B} = \mathbf{A}^{-1}$$

Minimal Number of Samples

Section 1: Deriving the Lower Bound

Minimal Number of Samples: *Observation One*

An Initial Lower Bound

Let $\mathbf{A}$ have M rows and $\boldsymbol{x}^\star$ be the unique solution of the ℓ_0-norm minimization. Then, $\|\boldsymbol{x}^\star\|_0 \leq M$

Proof:

Remember the ℓ_0-norm minimization algorithm

$$\min \|\boldsymbol{z}\|_0 \text{ subject to } \mathbf{A}\boldsymbol{z} = \boldsymbol{y}$$

Assume $\boldsymbol{x}^\star$ has $s > M$ nonzero entries. Since $\boldsymbol{x}^\star$ is solution,

$$\mathbf{A}\boldsymbol{x}^\star = \boldsymbol{y}$$

Minimal Number of Samples: *Observation One*

An Initial Lower Bound

Let $\mathbf{A}$ have M rows and $\boldsymbol{x}^\star$ be the unique solution of the ℓ_0-norm minimization. Then, $\|\boldsymbol{x}^\star\|_0 \leq M$

Proof:

Let $\mathcal{S} = \mathrm{Supp}\,(\boldsymbol{x}^\star)$*. Then, we can write*

$$\mathbf{A}_\mathcal{S}\boldsymbol{x}^\star_\mathcal{S} = \mathbf{A}\boldsymbol{x}^\star = \boldsymbol{y}$$

where $\mathbf{A}_\mathcal{S} \in \mathbb{R}^{M\times s}$*. Since* $\mathbf{A}_\mathcal{S}$ *is fat, we know that*

$$\ker \mathbf{A}_\mathcal{S} \neq \{\mathbf{0}\}$$

Minimal Number of Samples: *Observation One*

An Initial Lower Bound

Let $\mathbf{A}$ *have* M *rows and* $\boldsymbol{x}^\star$ *be the unique solution of the* ℓ_0*-norm minimization. Then,* $\|\boldsymbol{x}^\star\|_0 \leq M$

Proof:

Let $\ker \mathbf{A}_{\mathcal{S}} \ni \boldsymbol{u} \neq \mathbf{0}$*, and define* $\boldsymbol{x}^\sharp$ *to be*

$$\boldsymbol{x}^\star_{\mathcal{S}} + \boldsymbol{u} \text{ at entries of } \mathcal{S} \text{ and zero elsewhere} \equiv \boldsymbol{x}^\sharp_{\mathcal{S}} = \boldsymbol{x}^\star_{\mathcal{S}} + \boldsymbol{u}$$

Clearly, we have

$$\mathrm{Supp}(\boldsymbol{x}^\sharp) \subseteq \mathrm{Supp}\left(\boldsymbol{x}^\star\right) \rightsquigarrow \|\boldsymbol{x}^\sharp\|_0 \leq \|\boldsymbol{x}^\star\|_0$$

Minimal Number of Samples: *Observation One*

An Initial Lower Bound

Let $\mathbf{A}$ have M rows and $\boldsymbol{x}^\star$ be the unique solution of the ℓ_0-norm minimization. Then, $\|\boldsymbol{x}^\star\|_0 \leq M$

Proof:

Moreover, we can write

$$\begin{aligned}\mathbf{A}\boldsymbol{x}^\sharp = \mathbf{A}_{\mathcal{S}}\boldsymbol{x}^\sharp_{\mathcal{S}} &= \mathbf{A}_{\mathcal{S}}\boldsymbol{x}^\star_{\mathcal{S}} + \underbrace{\mathbf{A}_{\mathcal{S}}\boldsymbol{u}}_{\mathbf{0}} \\ &= \mathbf{A}_{\mathcal{S}}\boldsymbol{x}^\star_{\mathcal{S}} \\ &= \boldsymbol{y}\end{aligned}$$

Minimal Number of Samples: *Observation One*

An Initial Lower Bound

Let $\mathbf{A}$ have M rows and $\boldsymbol{x}^\star$ be the unique solution of the ℓ_0-norm minimization. Then, $\|\boldsymbol{x}^\star\|_0 \leq M$

Proof:

This means $\boldsymbol{x}^\sharp$ is either sparser than, or as sparse as $\boldsymbol{x}^\star$, i.e.,

$\boldsymbol{x}^\star$ *with* $\|\boldsymbol{x}^\star\|_0 > M$ *cannot be unique solution*

Or equivalently

If $\boldsymbol{x}^\star$ *is the unique solution; then,* $\|\boldsymbol{x}^\star\|_0 \leq M$

Minimal Number of Samples: *Observation One*

From observation one, we could conclude that

> *The minimal number of samples is larger than or equal to* s

In other words,

> *With $M < s$ samples, we cannot recover a sparse signal*

Minimal Number of Samples

Section 2: Deriving the Upper Bound

Minimal Number of Samples: *Observation Two*

An Upper Bound for Single Sparse Recovery

Let signal $\mathbf{x} \in \mathbb{R}^N$ be s-sparse. Then, there exists a matrix $\mathbf{A}$ with

$$M \geq s + 1$$

rows, such that $\mathbf{x}$ is recovered uniquely from samples

$$\mathbf{y} = \mathbf{A}\mathbf{x}$$

via the ℓ_0-norm minimization algorithm

Proof: *See page 53 of the text book*

Minimal Number of Samples: *Observation Two*

What does the second finding say?

Let's focus on a particular s-sparse signal $\boldsymbol{x} \in \mathbb{R}^N$; then,

- *We design a sampling matrix $\mathbf{A} \in \mathbb{R}^{(s+1)\times N}$ for signal $\boldsymbol{x}$*
- *We sample $\boldsymbol{x}$ as $\boldsymbol{y} = \mathbf{A}\boldsymbol{x}$*
- *We give the samples to ℓ_0-norm minimization algorithm*

We are sure the algorithm recovers the considered $\boldsymbol{x}$ uniquely

Which part of this finding is impractical?

The sampling matrix depends on the signal!

Minimal Number of Samples: *Observation Two*

Could you make it more clear?

Assume we design $\mathbf{A}$ *with* $s+1$ *rows to sample s-sparse signal* $\boldsymbol{x}$*, such that*

$$\boldsymbol{x} \text{ is recovered uniquely from the samples}$$

Then, there is no guarantee that if we use $\mathbf{A}$ *to sample another s-sparse signal*

$$\boldsymbol{z} \neq \boldsymbol{x}$$

we can still uniquely recover $\boldsymbol{z}$ *from the samples* $\hat{\mathbf{y}} = \mathbf{A}\boldsymbol{z}$*!*

Minimal Number of Samples: *Observation Two*

What kind of sampling matrix are we interested in?

We are interested in a sampling matrix which
guarantees the unique recovery of all s-sparse signals

We now try to construct a matrix with this property

Minimal Number of Samples: *Observation Three*

Assume we have distinct real numbers $0 < t_1 < \ldots < t_N$

$$\mathbf{A} = \begin{bmatrix} t_1^0 & \ldots & t_N^0 \\ t_1^1 & \ldots & t_N^1 \\ \vdots & & \vdots \\ t_1^{2s-1} & \ldots & t_N^{2s-1} \end{bmatrix} = \begin{bmatrix} 1 & \ldots & 1 \\ t_1 & \ldots & t_N \\ \vdots & & \vdots \\ t_1^{2s-1} & \ldots & t_N^{2s-1} \end{bmatrix}$$

For any $\mathcal{S} \subset \{1, \ldots, N\}$ *of size*

$$|\mathcal{S}| = \text{ \# of elements in } \mathcal{S} = 2s$$

the column $\mathcal{S}$*-submatrix of* $\mathbf{A}$ *is invertible*

See Appendix A of the textbook

Minimal Number of Samples: *Observation Three*

As a result for any $\mathcal{S}$ of size $2s$, we can write

$$\mathbf{A}_{\mathcal{S}} \text{ is invertible} \rightsquigarrow \mathbf{A}_{\mathcal{S}} \text{ is injective} \rightsquigarrow \ker \mathbf{A}_{\mathcal{S}} = \{\mathbf{0}\}$$

Now assume that we sample s-sparse signal $\boldsymbol{x}$ with $\mathbf{A}$

$$\boldsymbol{y} = \mathbf{A}\boldsymbol{x}$$

We want to make sure that there exists no $\boldsymbol{x}^\sharp$, such that

$$\boldsymbol{y} = \mathbf{A}\boldsymbol{x}^\sharp$$

and

$$\|\boldsymbol{x}^\sharp\|_0 \leq s$$

Minimal Number of Samples: *Observation Three*

Opposite Assumption: Assume there is a $\mathbf{x}^\sharp$; then, we have

$$\mathbf{A}\left(\mathbf{x} - \mathbf{x}^\sharp\right) = \mathbf{y} - \mathbf{y} = \mathbf{0}$$

Since $\|\mathbf{x}^\sharp\|_0 \leq \|\mathbf{x}\|_0 \leq s$, we could say

The number of non-zero entries in $\mathbf{x} - \mathbf{x}^\sharp$ is less than $2s$

What is the worst case?

$$\mathbf{x} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \text{ and } \mathbf{x}^\sharp = \begin{bmatrix} 0 \\ 3 \\ 3 \\ 0 \\ 0 \end{bmatrix} \rightsquigarrow \mathbf{x} - \mathbf{x}^\sharp = \begin{bmatrix} 1 \\ -3 \\ -3 \\ 0 \\ 1 \end{bmatrix}$$

Minimal Number of Samples: *Observation Three*

$$\boxed{\textit{The number of non-zero entries in } \boldsymbol{x} - \boldsymbol{x}^\sharp \textit{ is at most } 2s}$$

Therefore, there is a $\mathcal{S}$ of size $2s$ such that

$$\mathbf{A}_{\mathcal{S}} \left(\boldsymbol{x} - \boldsymbol{x}^\sharp\right)_{\mathcal{S}} = \mathbf{A} \left(\boldsymbol{x} - \boldsymbol{x}^\sharp\right) = \mathbf{0}$$

This means that

$$\boldsymbol{x}_{\mathcal{S}} - \boldsymbol{x}^\sharp_{\mathcal{S}} \in \ker \mathbf{A}_{\mathcal{S}}$$

Oops! But, $\ker \mathbf{A}_{\mathcal{S}} = \{\mathbf{0}\}$. *This means*

$$\boldsymbol{x} - \boldsymbol{x}^\sharp = \mathbf{0} \rightsquigarrow \boldsymbol{x} = \boldsymbol{x}^\sharp$$

Minimal Number of Samples: *Observation Three*

An Upper Bound for Universal Sparse Recovery

There exists a sampling matrix $\mathbf{A} \in \mathbb{R}^{M \times N}$ *with*

$$M \geq 2s$$

by which any s*-sparse signal* $\boldsymbol{x} \in \mathbb{R}^N$ *is recovered uniquely from*

$$\boldsymbol{y} = \mathbf{A}\boldsymbol{x}$$

via the ℓ_0*-norm minimization algorithm*

Proof: *We just did it by contradiction! We even know what is* $\mathbf{A}$

Minimal Number of Samples: *Observation Three*

From observation three, we could conclude that

$\boxed{\textit{The minimal number of samples is smaller than or equal to } 2s}$

In other words,

$\boxed{\textit{With } M \geq 2s \textit{ samples, we definitely recover an } s\textit{-sparse signal}}$

Summary

In a nutshell, we learned that ...

- *The optimal approach for sparse recovery is*

$$\ell_0\text{-norm minimization}$$

- *Minimum samples, we need to do unique sparse recovery reads*

$$s \leq M^\star \leq 2s$$

But can we use these results in practice?

$$\boxed{\textit{No! We explain why it is so in the next sections}}$$

Complexity of ℓ_0-Norm Minimization

Direct Solution to ℓ_0-Norm Minimization

ℓ_0-Norm Minimization

$$\min \|\boldsymbol{z}\|_0 \textit{ subject to } \mathbf{A}\boldsymbol{z} = \boldsymbol{y}$$

What is the direct approach to solve this problem?

We need to find all vectors $\boldsymbol{z}$ *that satisfy*

$$\mathbf{A}\boldsymbol{z} = \boldsymbol{y}$$

and then choose the solution with minimum ℓ_0*-norm!*

Direct Solution to ℓ_0-Norm Minimization

ℓ_0-Norm Minimization

$$\min \|\boldsymbol{z}\|_0 \text{ subject to } \mathbf{A}\boldsymbol{z} = \boldsymbol{y}$$

What is the direct approach to solve this problem?

This is however a search with infinite feasible points!

This is not possible to implement!

This approach seems to be infeasible

Direct Solution to ℓ_0-Norm Minimization

ℓ_0-Norm Minimization

$$\min \|\boldsymbol{z}\|_0 \text{ subject to } \mathbf{A}\boldsymbol{z} = \boldsymbol{y}$$

Is there an alternative approach?

Well, we could go for the original sparse recovery problem, i.e.,

$$\mathbf{A}\boldsymbol{z} = \boldsymbol{y} \text{ subject to } \|\boldsymbol{z}\|_0 \leq s$$

for some known s

Direct Solution to ℓ_0-Norm Minimization

ℓ_0-Norm Minimization: *Alternative Problem*

$$\mathbf{A}\boldsymbol{z} = \boldsymbol{y} \text{ subject to } \|\boldsymbol{z}\|_0 \leq s$$

But how can we solve this alternative problem?

We could set $\|\boldsymbol{z}\|_0 = s$: Consider all possible supports

$$\# \text{ of possible supports } = \binom{N}{s}$$

For each support, we solve $\mathbf{A}\boldsymbol{z} = \boldsymbol{y}$

Direct Solution to ℓ_0-Norm Minimization

ℓ_0-Norm Minimization: *Alternative Problem*

$$\mathbf{A}\boldsymbol{z} = \boldsymbol{y} \text{ subject to } \|\boldsymbol{z}\|_0 \leq s$$

But how can we solve this alternative problem?

Let $\mathcal{S} \subseteq \{1, \ldots, N\}$ be a support with $|\mathcal{S}| = s$; then,

$$\mathbf{A}\boldsymbol{z} = \boldsymbol{y} \rightsquigarrow \mathbf{A}_{\mathcal{S}}\boldsymbol{z}_{\mathcal{S}} = \boldsymbol{y}$$

Depending on $\mathbf{A}_{\mathcal{S}}$, this is determined, under- or overdetermined

Direct Solution to ℓ_0-Norm Minimization

ℓ_0-Norm Minimization: *Alternative Problem*

$$\mathbf{A}\boldsymbol{z} = \boldsymbol{y} \text{ subject to } \|\boldsymbol{z}\|_0 \leq s$$

But how can we solve this alternative problem?

$\mathbf{A}_{\mathcal{S}}$ *has* M *rows and* $s \leq M$ *columns*

If less than s *linearly independent rows are in* $\mathbf{A}_{\mathcal{S}}$

Then, the problem is underdetermined, but this is impossible

Direct Solution to ℓ_0-Norm Minimization

ℓ_0-Norm Minimization: *Alternative Problem*

$$\mathbf{A}\boldsymbol{z} = \boldsymbol{y} \text{ subject to } \|\boldsymbol{z}\|_0 \leq s$$

But how can we solve this alternative problem?

Why impossible? Remember that underdetermined means

We have multiple solutions to the sparse recovery problem

But we have a unique solution. This is hence impossible

Direct Solution to ℓ_0-Norm Minimization

ℓ_0-Norm Minimization: *Alternative Problem*

$$\mathbf{A}\boldsymbol{z} = \boldsymbol{y} \text{ subject to } \|\boldsymbol{z}\|_0 \leq s$$

But how can we solve this alternative problem?

$\mathbf{A}_{\mathcal{S}}$ has M rows and $s \leq M$ columns

With s independent rows and no other conflicting equations

The problem is determined with a single solution

Direct Solution to ℓ_0-Norm Minimization

ℓ_0-Norm Minimization: *Alternative Problem*

$$\mathbf{A}\boldsymbol{z} = \boldsymbol{y} \text{ subject to } \|\boldsymbol{z}\|_0 \leq s$$

But how can we solve this alternative problem?

$\mathbf{A}_{\mathcal{S}}$ *has* M *rows and* $s \leq M$ *columns*

With s *independent rows and conflicting equations*

The problem is overdetermined with no solution

Direct Solution to ℓ_0-Norm Minimization

ℓ_0-Norm Minimization: *Alternative Problem*

$$\mathbf{A}\boldsymbol{z} = \boldsymbol{y} \text{ subject to } \|\boldsymbol{z}\|_0 \leq s$$

But how can we solve this alternative problem?

Once we are over with $\|\boldsymbol{z}\|_0 = s$*, we go for* $\|\boldsymbol{z}\|_0 = s - 1, \ldots,$

Not really needed, since $(s-1)$*-sparse vector is also* s*-sparse*

We could limit the search to supports with s *elements*

Direct Solution to ℓ_0-Norm Minimization

ℓ_0-Norm Minimization: *Alternative Problem*

$$\mathbf{A}\boldsymbol{z} = \boldsymbol{y} \text{ subject to } \|\boldsymbol{z}\|_0 \leq s$$

But how can we solve this alternative problem?

Since the solution is unique, we could say

There is only one support, for which the problem is determined

So, we keep trying till we end up with the determined problem

Direct Solution to ℓ_0-Norm Minimization

How long does it take to find this solution?

Well, it depends!

- *If we are lucky!*
  - *We find the solution in early steps*
- *If we are not lucky!*
  - *We find the solution in final steps*

What does the statistics say?

We are half of the times lucky, half of the time unlucky

Direct Solution to ℓ_0-Norm Minimization

Then, what would be a typical search time?

Well! Let's say we do in average half of the searches:

- *We should check C cases, where*

$$C = \frac{1}{2}\binom{N}{s}$$

- *Say each check takes T sec, we have*

$$\text{Search time } = CT = \frac{1}{2}\binom{N}{s} T \text{ sec}$$

Direct Solution to ℓ_0-Norm Minimization

What would be the value for a typical application?

Consider a typical MRI:

- *We have around* $N = 10^7$ *pixels*
- *Say the MRI scan is* 1% *sparse, this mean*

$$s = 0.01 \times 10^7 = 10^5$$

- *So, we could say*

$$C = \frac{1}{2}\binom{N}{s} \geq \frac{1}{2}\left(\frac{N}{s}\right)^{s} = \frac{1}{2} \times 100^{10^5} = \frac{1}{2} \times 10^{200000}$$

Direct Solution to ℓ_0-Norm Minimization

What would be the value for a typical application?

Consider a typical MRI:

- *Say each search takes only* $T = 10^{-10}$ *sec; then,*

$$\text{Search time} = \frac{1}{2} \times 10^{199990} \text{ sec}$$

- *This is more than* 10^{199979} *Centenaries!*

Forget about it!

ℓ_0*-norm Minimization is not a possible to implement!*

NP-hardness of ℓ_0-Norm Minimization

One might say: The direct approach is not feasible, but

How could we conclude that there is not an efficient way?!

Well, as usual

We need to go through some definitions to answer that

Some Definitions: *P-Problem*

Polynomial Computational Complexity

A polynomial time algorithm is an algorithm whose number of operations is a polynomial function of input dimension

Example: Finding the minimum entry of a vector

Say the input is a vector of length N

- *Compare the first two entries and find the minimum*
- *Compare minimum with next one, and find new minimum*
- *Keep repeating till the last entry*

Some Definitions: *P-Problem*

Polynomial Computational Complexity

A polynomial time algorithm is an algorithm whose number of operations is a polynomial function of input dimension

Example: Finding the minimum entry of a vector

We need to do $N-1$ comparisons in general

$$\# \text{ of operations } = N - 1$$

which is linear in N

Some Definitions: *P-Problem*

P-Problems

The class of P-problems is the set of all decision problems whose solution is given by a polynomial time algorithm

Why decision problems?

Roughly speaking, when it come to implementation

We can reformulate most problem with decision problems

You can ignore the phrase decision for the moment

Some Definitions: *NP-Problem*

NP-Problems

The class of nondeterministic polynomial time problems is the set of all decision problems for which there is a polynomial time algorithm to certify the solution

What does that mean?

Even if we cannot solve the problem in polynomial time,

When a solution is given, we can check it in polynomial time

Clearly, we can say

$$\{\textit{P-Problems}\} \subseteq \{\textit{NP-Problems}\}$$

Some Definitions: *NP-Problem*

Example: Exact Cover by Subsets of Three

We are given with a set

$$\{\mathcal{I}_1, \ldots, \mathcal{I}_N\}$$

Each $\mathcal{I}_n$ contains three integers from $\{1, \ldots, m\}$

We look for J non-overlapping subsets $\mathcal{I}_{i_1}, \ldots, \mathcal{I}_{i_J}$, such that

$$\mathcal{I}_{i_1} \cup \ldots \cup \mathcal{I}_{i_J} = \{1, \ldots, m\}$$

Some Definitions: *NP-Problem*

Example: Exact Cover by Subsets of Three

If someone claims that $\mathcal{I}_{i_1}, \ldots, \mathcal{I}_{i_J}$ is a solution

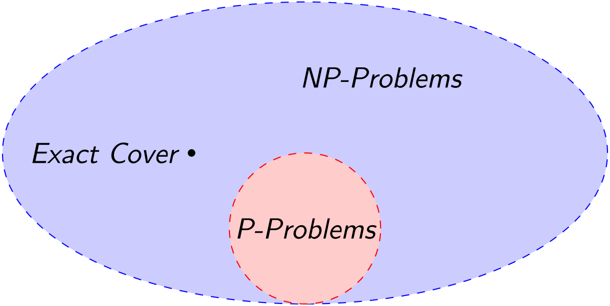
We can check the validity of this claim in polynomial time

Therefore, we could say

$$\textit{Exact Cover by Subsets of Three} \in \{\textit{NP-Problems}\}$$

Some Definitions: *NP-Problem*

Where does Exact Cover by Subsets of Three locate?



Some Definitions: *NP-Hardness*

NP-Hard Problems

The class of NP-hard problems is the set of all problems whose solving algorithm is converted to the solving algorithm of any NP-problem in polynomial time

What does that mean?

First of all, note that

We don't know any polynomial time algorithm

which can solve an NP-hard problem

Some Definitions: *NP-Hardness*

NP-Hard Problems

The class of NP-hard problems is the set of all problems whose solving algorithm is converted to the solving algorithm of any NP-problem in polynomial time

Assume there is an oracle machine

$$\textit{NP-Hard} \longrightarrow \boxed{\textit{Oracle}} \longrightarrow \textit{Solution}$$

This machine solves only one NP-hard problem

Some Definitions: *NP-Hardness*

NP-Hard Problems

The class of NP-hard problems is the set of all problems whose solving algorithm is converted to the solving algorithm of any NP-problem in polynomial time

Then we can make a polynomial time algorithm out of it

NP → *Polynomial Algorithm* → *Oracle* → *Solution*

for any NP-problem

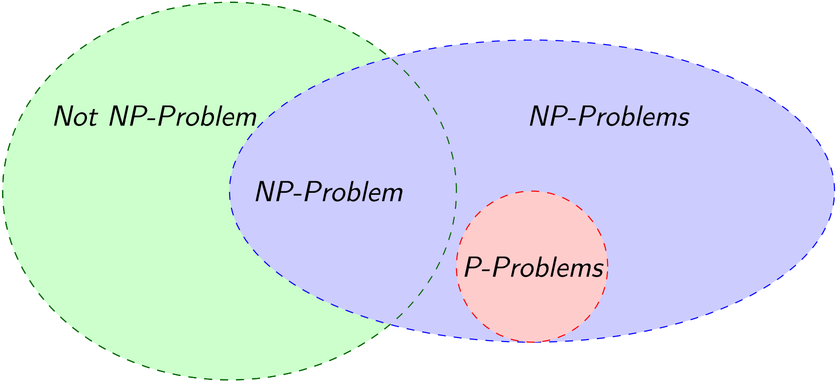
Some Definitions: *NP-Hardness*

We could say

NP-hard problems are core hard problems

If one day we solve one, we solve all NP-problems

Some Definitions: *NP-Hardness*



Some Definitions: *NP-Complete*

But what if someone claims that

He/she got a solution to an NP-hard problem?

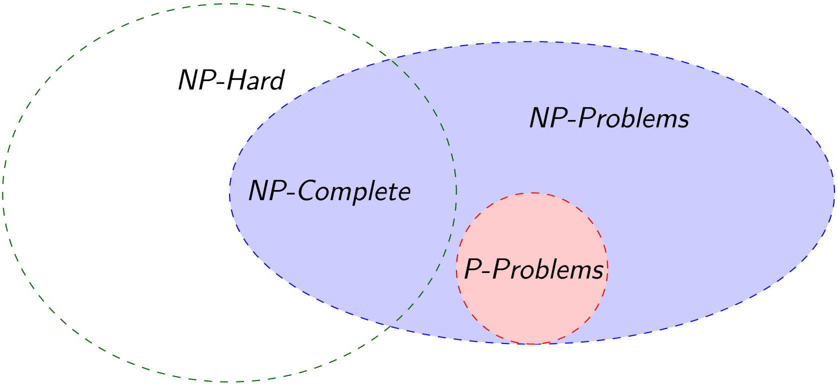
Of course, we need to be able to verify the claim

Better if the problem is an NP-problem

NP-Complete Problems

The class of NP-complete problems is the set of all NP-hard problems which are NP-problems as well

Some Definitions: *NP-Complete*



Some Definitions: *NP-Complete*

We could say

> *NP-complete problems are atom hard problems*

If one day we come up with a solution,

- *We can validate the solution*
- *We can solve all NP-problems*

> *Examples of NP-complete problems*
> - *The salesman problem*
> - *Exact cover by subsets of three*

NP-Hardness of Optimal Recovery

What is the use of all of these?

> *We use these definitions to check whether*
>
> *A problem is feasible to solve at the moment or not*

How can we do it?

> *If a given problem describes an NP-complete problem*
>
> *Our current tools cannot solve the given problem!*

NP-Hardness of Optimal Recovery

Let's get back to exact cover by subsets of three

$\mathcal{I}_1, \ldots, \mathcal{I}_N$ *contain three integers from* $\{1, \ldots, m\}$

We want to cover $\{1, \ldots, m\}$ *by* J *non-overlapping of them*

Example: Assume $m = 9$ *and* $N = 4$ *with*

$$\mathcal{I}_1 = \{1, 2, 3\} \ , \ \mathcal{I}_2 = \{3, 4, 5\} \ , \ \mathcal{I}_3 = \{4, 5, 6\} \ , \ \mathcal{I}_4 = \{7, 8, 9\}$$

Then, the solution has $J = 3$ *subsets which are*

$$\mathcal{I}_1, \mathcal{I}_3, \mathcal{I}_4$$

NP-Hardness of Optimal Recovery

Let's get back to exact cover by subsets of three

$\mathcal{I}_1, \ldots, \mathcal{I}_N$ *contain three integers from* $\{1, \ldots, m\}$

We want to cover $\{1, \ldots, m\}$ *by* J *non-overlapping of them*

Let us show each $\mathcal{I}_n$ *by a vector* $\boldsymbol{a}_n \in \{0, 1\}^m$*, e.g.,*

$$\mathcal{I}_1 = \{1, 2, 3\} \rightsquigarrow \boldsymbol{a}_1 = [1, 1, 1, 0, \ldots, 0]^\mathsf{T}$$

and construct the matrix $\mathbf{A}$ *as*

$$\mathbf{A} = [\boldsymbol{a}_1, \ldots, \boldsymbol{a}_N]$$

NP-Hardness of Optimal Recovery

Let's get back to exact cover by subsets of three

$\mathcal{I}_1, \ldots, \mathcal{I}_N$ *contain three integers from* $\{1, \ldots, m\}$

We want to cover $\{1, \ldots, m\}$ *by* J *non-overlapping of them*

Now let $\boldsymbol{y}$ *be the vector of* m *ones*

$$\boldsymbol{y} = [\underbrace{1, \ldots, 1}_{m \text{ times}}]^{\mathsf{T}}$$

NP-Hardness of Optimal Recovery

Let's get back to exact cover by subsets of three

$\mathcal{I}_1, \ldots, \mathcal{I}_N$ *contain three integers from* $\{1, \ldots, m\}$

We want to cover $\{1, \ldots, m\}$ *by* J *non-overlapping of them*

Collection of J *subsets could be shown by* $\boldsymbol{x} \in \{0, 1\}^N$*, e.g.,*

$$\mathcal{I}_1, \mathcal{I}_3, \mathcal{I}_4 \rightsquigarrow \boldsymbol{x} = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 1 \end{bmatrix}$$

NP-Hardness of Optimal Recovery

Let's get back to exact cover by subsets of three

> $\mathcal{I}_1, \ldots, \mathcal{I}_N$ *contain three integers from* $\{1, \ldots, m\}$
>
> *We want to cover* $\{1, \ldots, m\}$ *by* J *non-overlapping of them*

Since we aim to cover $\{1, \ldots, m\}$*, we look for an* $\boldsymbol{x}$

$$\mathbf{A}\boldsymbol{x} = \boldsymbol{y}$$

$\boldsymbol{x}$ *is an sparse vector which contains only* $J = m/3$ *non-zero entry*

NP-Hardness of Optimal Recovery

Let's get back to exact cover by subsets of three

$\mathcal{I}_1, \dots, \mathcal{I}_N$ *contain three integers from* $\{1, \dots, m\}$

We want to cover $\{1, \dots, m\}$ *by* J *non-overlapping of them*

$\boldsymbol{x}$ *is the sparsest possible solution*

Any sparser $\boldsymbol{x}$ *would not cover* $\{1, \dots, m\}$

We could hence say

$$\boldsymbol{x} = \operatorname{argmin} \|\boldsymbol{z}\|_0 \ \ \text{subject to } \mathbf{A}\boldsymbol{z} = \boldsymbol{y}$$

NP-Hardness of Optimal Recovery

What can we now conclude?

$\boxed{\ell_0 \text{ minimization reduces to an NP-complete problem}}$

Thus, ℓ_0 minimization is an NP-hard problem

$\boxed{\text{We do not know any algorithm which solves it in a feasible time}}$

This is why we go for other algorithms!

Final Points

Summary

- *The optimal approach for sparse recovery is*

  ℓ_0-norm minimization

- *For unique sparse recovery via the optimal approach at least*

  $$s \leq M^{\star} \leq 2s$$

  samples should be collected

- *ℓ_0-norm minimization is an NP-hard problem*

  It cannot be implemented in practice

What We Learn Next?

- *We learn sub-optimal algorithms which can be implemented*
- *Using these algorithms,*

The minimum number of samples changes

Which Parts of Textbooks?

$\boxed{\text{We are over with this part}}$

I would suggest to go over the text book

A Mathematical Introduction to Compressive Sensing
S. Foucar and H. Rauhut, Book, 2013

and study the following part:

- *Chapter 2: Sections 2.2 and 2.3, Pages 48–59*